



Fondamenti d'Informatica: stringa e linguaggio

Barbara Re, Phd



Agenda

- ▶ **Introdurremo ...**
 - ▶ ... la nozione di stringa e linguaggio
 - ▶ ... il ruolo che stringhe e linguaggi hanno per rappresentare l'informazione

Alfabeti e Stringe

- ▶ I processi di calcolo considerano **stringhe di simboli** (detti caratteri) scelti in un insieme finito e non vuoto, chiamato **Alfabeto**
es: $\{0, 1\}$, $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, ... $\{a, b, c, d\}$
- ▶ Una sequenza finita di simboli di un dato alfabeto A viene chiamata **stringa** o **parola** su A
- ▶ La stringa, formata dalla sequenza di simboli a_1, a_2, \dots, a_n viene usualmente denotata $a_1 a_2 \dots a_n$
- ▶ Una **stringa costituita da nessun simbolo**, detta vuota è denotata da λ
- ▶ La **lunghezza di una stringa** α , denotata come $l(\alpha)$ è il numero dei simboli che la compongono
 - ▶ $\alpha = a_1 a_2 \dots a_n - l(\alpha) = n$

Concatenazione di stringhe

- ▶ Date due stringhe α, β se ne può formare un'altra in cui α è seguita da β , essa è denotata come $\alpha\beta$ e chiamata la **concatenazione** di α, β
- ▶ La notazione α^i è usata per la stringa ottenuta concatenando i copie della stringa α quando i è intero positivo ($\alpha^0 = \lambda$, $\alpha^1 = \alpha$, $\alpha^2 = \alpha\alpha$, $\alpha^3 = \alpha\alpha\alpha$, ...)

Esempio: Dato l'alfabeto $\{a, b\}$ a cosa fa riferimento l'insieme $\{a^n b^n \mid n \geq 1\}$?

E' il linguaggio composto da tutte le stringhe costituite dalla concatenazione di un certo numero di a , seguito dalla concatenazione dello stesso numero di b

Sottostringhe, prefissi, suffissi

- ▶ Una stringa α è una **sottostringa** di β se $\beta = \gamma\alpha\rho$ per qualche scelta di stringhe $\gamma\rho$
- ▶ Una sottostringa α di una stringa β si chiama **prefisso** di β se $\beta = \alpha\rho$ per qualche stringa ρ (dunque per $\gamma = \lambda$)
 - ▶ α è un **prefisso proprio** se $\rho \neq \lambda$
- ▶ Una sottostringa α di una stringa β è un **suffisso** di β se $\beta = \gamma\alpha$ per qualche stringa γ (dunque per $\rho = \lambda$)
 - ▶ α è un **suffisso proprio** se $\gamma \neq \lambda$
- ▶ Se $\alpha = a_1 a_2 \dots a_n$, allora $a_n \dots a_2 a_1$ è chiamata **l'inversa** di α e denotata α^{rev}

A^* e A^+

- ▶ A^* è l'insieme di tutte le stringhe sull'alfabeto A (anche detto linguaggio), dove $*$ prende il nome di **iterazione** o stella di Kleene
- ▶ A^+ denota l'insieme $A^* - \{\lambda\}$ delle stringhe non vuote su A

$$A^* = \bigcup_{h=0}^{\infty} A^h$$

$$A^+ = \bigcup_{h=1}^{\infty} A^h$$

Ordinamento

- ▶ Un alfabeto A , come **insieme finito di simboli**, si può ordinare in diverse modalità
 - ▶ **Ordinamento totale** $<$ significa assumere per $a_1 a_2 \dots a_n$ che $a_1 < a_2 < \dots < a_n$
- ▶ Una strategia di ordinamento frequentemente usata è quella **lessicografica**

Def. Sia A un alfabeto (ordinato da qualche relazione $<$) e siano α e β stringhe in A^* . Si dice che α è **lessicograficamente minore** di β , $\alpha < \beta$, o equivalentemente β è **lessicograficamente maggiore** di α , $\beta > \alpha$ se vale uno dei due casi:

- ▶ α è prefisso proprio di β
- ▶ α ha un prefisso γa_1 , β ha un prefisso γa_2 , per lo stesso $\gamma \in A^*$, con $a_1, a_2 \in A$ e $a_1 < a_2$ in A

I numeri primi

- ▶ Consideriamo i numeri naturali intesi come parole su $A = \{0, 1, 2, 3, \dots, 9\}$ rispetto alla loro notazione in base 10
- ▶ N è primo se?
 - ▶ $N \geq 2$
 - ▶ Gli unici divisori di N sono 1 e N
 - ▶ Ogni $N \geq 2$ si decompone in modo unico nel prodotto di fattori primi
- ▶ Nascono così due classi di problemi: entrambi con input $N \geq 2$
- ▶ Ci si chiede se N è primo o no
- ▶ Ci si chiede di decomporre N nei suoi fattori primi

Si vuole riconoscere tra le stringhe su A quelle che corrispondono ai numeri primi che di fatto è **un sottoinsieme di A**

Si vuole calcolare per ogni N quelle stringhe che rappresentano i numeri primi che dividono N , **computando una funzione che da parole su A genera nuove sequenze di parole su A**

Generalizzando

- ▶ Dato un alfabeto A , un sottoinsieme L di A^* si chiama linguaggio formale, o più semplicemente linguaggio o anche problema
- ▶ Si chiama linguaggio vuoto, e lo si indica con Λ , il linguaggio che non contiene stringa alcuna
- ▶ L'alfabeto fa riferimento alla sua natura d'insieme di parole
- ▶ La questione computazionale che L propone fa riferimento alla possibilità di **riconoscere** le stringhe su A che stanno in L e esclude le altre e anche alla computazione

$\Lambda \neq \lambda$

Riconoscere -> **Compilatore**

- ▶ Tra i problemi più basilari che debbono essere affrontati nell'informatica vi è quello di riconoscere se una stringa appartenga o meno ad un determinato linguaggio, opportunamente definito.
- ▶ Ad esempio, quando un **compilatore** analizza un programma in **linguaggio C o Pascal** deve verificare che tale programma sia sintatticamente corretto, o, in altri termini, che la stringa costituita dal programma stesso appartenga al linguaggio dei programmi C o Pascal sintatticamente corretti

Operazioni su linguaggi

- ▶ Dati due linguaggi L_1 ed L_2 si possono definire su essi varie operazioni:
 - ▶ Operazioni binarie di **intersezione**, **unione** e **concatenazione**
 - ▶ Operazioni unarie di complementazione ed iterazione
- ▶ L' **intersezione** di due linguaggi L_1 e L_2 è il linguaggio $L_1 \cap L_2$ costituito dalle parole presenti sia in L_1 che di L_2 , cioè

$$L_1 \cap L_2 = \{x \in A^* \mid x \in L_1 \text{ and } x \in L_2\}$$

- ▶ L' **unione di due linguaggi** L_1 e L_2 è il linguaggio $L_1 \cup L_2$ costituito dalle parole appartenenti ad almeno uno fra L_1 ed L_2 , cioè

$$L_1 \cup L_2 = \{x \in A^* \mid x \in L_1 \text{ or } x \in L_2\}$$

$$L_1 \cap \Lambda = \Lambda \text{ e } L_1 \cup \Lambda = L_1$$

Complemento e concatenazione

- ▶ Il **complemento di un linguaggio** L_1 è il linguaggio $\bar{L} = A^* - L_1$ costituito dalle parole appartenenti a A^* ma non ad L_1 , cioè

$$\bar{L} = \{x \in A^* \mid x \notin L_1\}.$$

- ▶ La **concatenazione** (o prodotto) di due linguaggi L_1 e L_2 è il linguaggio $L_1 \circ L_2$ delle parole costituite dalla concatenazione di una stringa di L_1 e di una stringa di L_2 , cioè

$$L_1 \circ L_2 = \{x \in A^* \mid \exists y_1 \in L_1 \exists y_2 \in L_2 (x = y_1 \circ y_2)\}.$$

$$L \circ \{\lambda\} = \{\lambda\} \circ L = L, \text{ e che } L \circ \Lambda = \Lambda \circ L = \Lambda$$

Potenza

- ▶ La **potenza** L^h di un linguaggio è definita come

$$L^h = L \circ L^{h-1}, h \geq 1$$

con la convenzione secondo cui $L_0 = \{\lambda\}$

L^* e L^+

- ▶ **Il linguaggio L^* come di seguito definito** prende il nome di chiusura riflessiva del linguaggio L rispetto all'operazione di concatenazione, mentre l'operatore “ $*$ ” prende il nome di iterazione o stella di Kleene
- ▶ **L^+** denota la chiusura (non riflessiva)

$$L^* = \bigcup_{h=0}^{\infty} L^h$$

$$L^+ = \bigcup_{h=1}^{\infty} L^h$$

